



This is a repository copy of *Gaussian process upper confidence bounds in distributed point target tracking over wireless sensor networks*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/193228/>

Version: Accepted Version

Article:

Liu, X., Mihaylova, L. orcid.org/0000-0001-5856-2223, George, J. et al. (1 more author) (2023) Gaussian process upper confidence bounds in distributed point target tracking over wireless sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, 17 (1). pp. 295-310. ISSN 1932-4553

<https://doi.org/10.1109/JSTSP.2022.3223521>

© 2022 The Author(s). This is an author-produced version of a paper subsequently published in *IEEE Journal of Selected Topics in Signal Processing*. This version is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>).

Reuse

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

Gaussian Process Upper Confidence Bounds in Distributed Point Target Tracking over Wireless Sensor Networks

Xingchi Liu*, Lyudmila Mihaylova*, Jemin George[†], Tien Pham[‡]

*University of Sheffield, UK

[†]DEVCOM Army Research Laboratory, USA

[‡]MITRE Labs, USA

Email: xingchi.liu@sheffield.ac.uk l.s.mihaylova@sheffield.ac.uk jemin.george.civ@army.mil, tienpham@mitre.org

Abstract—Uncertainty quantification plays a key role in the development of autonomous systems, decision-making, and tracking over wireless sensor networks (WSNs). However, there is a need of providing uncertainty confidence bounds, especially for distributed machine learning-based tracking, dealing with different volumes of data collected by sensors. This paper aims to fill in this gap and proposes a distributed Gaussian process (DGP) approach for point target tracking and derives upper confidence bounds (UCBs) of the state estimates. A unique contribution of this paper includes the derived theoretical guarantees on the proposed approach and its maximum accuracy for tracking with and without clutter measurements. Particularly, the developed approaches with uncertainty bounds are generic and can provide trustworthy solutions with an increased level of reliability. A novel hybrid Bayesian filtering method is proposed to enhance the DGP approach by adopting a Poisson measurement likelihood model. The proposed approaches are validated over a WSN case study, where sensors have limited sensing ranges. Numerical results demonstrate the tracking accuracy and robustness of the proposed approaches. The derived UCBs constitute a tool for trustworthiness evaluation of DGP approaches. The simulation results reveal that the proposed UCBs successfully encompass the true target states with 88% and 42% higher probability in X and Y coordinates, respectively, when compared to the confidence interval-based method.

Index Terms—Distributed learning, target tracking, wireless sensor networks, Gaussian process methods, uncertainty quantification, upper confidence bounds, trustworthy solutions

I. INTRODUCTION

Target tracking in wireless sensor networks (WSNs) is a fundamental task for various applications including sea

We are grateful to the support for the UK Dstl and US MoD SIGNeTS project. Research was sponsored by the US Army Research Laboratory and the UK MOD University Defence Research Collaboration (UDRC) in Signal Processing, and was accomplished under Cooperative Agreement Number W911NF-20-2-0225. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the MOD, the U.S. Government or the U.K. Government. The U.S. Government and U.K. Government are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. We acknowledge the support from the EPSRC through EP/T013265/1 project NSF-EPSRC: “ShiRAS. Towards Safe and Reliable Autonomy in Sensor Driven Systems” and the support for ShiRAS by the USA National Science Foundation under Grant NSF ECCS 1903466. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

surveillance, autonomous vehicles, and traffic monitoring. The objective is to collect sensor measurements from one or multiple targets to estimate their current and future states [1]. However, the measurements may not only originate from the targets but also from the environmental interference (e.g. from ground and rain), which is referred to as the clutter [2]. To achieve reliable performance, a tracker should be able to distinguish between target measurements and clutter measurements, and also decide which measurement is associated with which target, which is called the data association problem [3].

To improve the tracking performance, numerous model-based approaches have been proposed including Kalman filter, extended Kalman filter [4], unscented Kalman filter [5], and particle filter [6], [7]. The posterior Cramér Rao Lower bound (PCRLB) for tracking can be calculated with different methods, e.g. as it calculated in [8]–[12].

However, these approaches rely on well-defined motion models, in particular, the target dynamics model and the sensor measurement model, which can be inaccurate when the target undergoes non-stationary evolution or mixed maneuvering behaviours. The multiple-model method [13] can capture complex behaviours by running a bank of elemental filters, each based on a unique model in the set and generating the overall estimates based on the results of these elemental filters. On the downside, this framework suffers from high computational complexity, and therefore, is not efficient when a large number of models are involved.

Tracking with multiple models can be achieved by Gaussian process (GP)-based model-free methods, which are powerful non-parametric machine learning inference methods [14]. Instead of tracking via motion models, GP-based methods can directly learn unknown functions which maps some system context information (input) to the target state (output). Particularly, GP regression can provide uncertainty quantification on the predictions. Ignoring this uncertainty can have disastrous consequences, especially when the output of such models is then fed into higher-level decision-making procedures. Recently, GP regression has been applied to solve both point target tracking [15]–[21] and extended target tracking [22]–[24] problems.

Most of the existing GP-based tracking methods assume

that the sensor measurements are collected in a centralized manner and both training and state estimation are made upon the aggregated measurements [25], [26]. However, due to the nature of distributed sensing systems, collecting all the measurements for GP training can bring a high communication cost, which is energy inefficient. In addition, considering a WSN with sheer amounts of sensors and measurements, the centralized tracking framework has inevitably reached an inherent bottleneck of scalability, which stems from the cubic computational complexity ($\mathcal{O}(N^3)$, where N is the number of measurements) of the standard GP regression due to the inversion and determinant calculations of the GP covariance matrix. Therefore, transferring the centralized-based approaches into distributed ones has become a popular choice, and multiple approaches including distributed consensus approaches [27] and message passing methods [28] have been proposed for target tracking. However, to the best of the authors' knowledge, there are rarely any studies about leveraging GP regression to solve the tracking and data association in a distributed way and provide theoretical performance analysis. Hence, in this paper, the distributed Gaussian process (DGP) framework is adopted to design a DGP-based tracking (DGPT) approach, which is able to learn the target motion online and solve the data association jointly in a data-driven way.

GP has also been studied in resource allocation problems under the edge computing framework to account for efficient decision-making and network state prediction [29], [30]. The presented DGPT approach can also rely on edge computing thanks to their distributed learning nature and since learning happens near the areas of data collection. The DGP framework has been implemented and tested on edge devices [31], and the whole edge learning process can be implemented in other ways as summarized in the survey [32].

The main contributions of this paper are summarized below:

- A DGPT approach is proposed for point target tracking under the assumption that the number of measurements follows a Poisson distribution. The proposed approach can leverage both temporal and spatial features to learn the hyperparameters of the DGP online, through a sliding window of measurements.
- To solve the data association problem, different weights are assigned to the measurements based on the marginal likelihood of local GPs and a weighted summation is calculated for DGP training. This method achieves efficient hyperparameter learning and state prediction. We justify that the complexity of the DGPT does not scale with the number of measurement, but only with the length of the sliding window and the number of active sensors.
- For the first time, this work derives theoretically upper confidence bounds (UCBs) for the state estimation error of the proposed DGPT. Numerical results demonstrate the superiority of the UCBs as compared to the confidence interval of the DGP model itself.
- With the knowledge learned from the DGP, a novel hybrid Bayesian filtering method is proposed to combine distributed machine learning and classical Bayesian inference. The designed tracker refines DGPT's state estimation by introducing a Poisson likelihood model,

which elegantly sidesteps the data association challenge.

- The performance of the proposed DGP and hybrid Bayesian filtering approaches are tested with challenging target trajectory scenarios - from uniform motions to highly maneuvering ones. Different levels of measurement noise and clutter rates are involved to evaluate the accuracy and robustness of the proposed approaches.

The remaining part of the paper is organized as follows. Section II reviews the related works. Section III introduces the fundamentals of GP regression and multiple DGP methods. Section IV describes the proposed DGPT approach followed by the theoretical analysis of the tracking error bound in Section V. Section VI describes the DGP-assisted hybrid Bayesian filtering approach. The simulation setup and results are presented in Section VII, and the conclusions are drawn in Section VIII. Appendices A, B, and C contain details about the theoretical derivations.

II. RELATED WORK

A wealth of approaches for improving the scalability of the standard GP method have been studied including centralized, distributed, and hybrid methods [33]. In [34], [35], sparse approximations of the original $N \times N$ covariance matrix of GP are obtained to summarize the dependence of the whole training data using M inducing points, which greatly reduces the computational complexity to $\mathcal{O}(NM^2)$. However, this type of methods requires all the data to make predictions and thus still works in a centralized manner.

The DGP which originates from the idea of divide-and-conquer [36], focuses on training local GPs (which are also referred to as local experts) based on subsets of the whole training data or based on the partitioning of the big state vector into state vectors with smaller dimensions [37]. After training the local experts, a family of aggregation methods [38]–[41] relying on the product of experts model, can be applied to aggregate the local knowledge together by multiplying local predictions, and then the overall prediction can be calculated. In addition, hybrid GP variants are studied to utilize a master GP expert to communicate with the local experts leading to a consistent posterior predictive distribution [42]. DGP has been applied to problems such as the received signal strength-based location fingerprinting map construction [43].

GP regression is also applied to solve the data association problem. In [44], GP priors are placed on the different generative processes and the associations are modelled via a latent association matrix and inference is carried out using an expectation-maximization algorithm. [45] extends GP-based data association into the non-stationary process where a different number of generative processes can be activated in different locations in the input space. GP has been also combined with state-space models to solve the tracking problem. In [46], one-dimensional temporal GP regression models are reformulated as linear-Gaussian state-space models, which can be solved exactly with classical Kalman filter. The state-space model representation is also used in spatial-temporal GPs [47] and non-Gaussian likelihood [48] to derive computationally efficient infinite-dimensional Kalman filtering and smoothing

Table I: A summary of recent works on GP-assisted tracking and localization

Target type	Ref	Data association	Centralized / Distributed	Data / Model-driven
Point target	[15]	No	Centralized	Data-driven
	[16]	No	Centralized	Data-driven (hybrid)
	[18]	No	Centralized	Data-driven (hybrid)
	[19]	No	Centralized	Data-driven (hybrid)
	[20]	No	Distributed	Date-driven
	[52]	No	Distributed	Model-driven
	[27]	Yes	Distributed	Model-driven
	[43]	No	Distributed	Data-driven
	[53]	No	Distributed	Data-driven
	[25]	No	Centralized	Data-driven (hybrid)
	[26]	Yes	Centralized	Data-driven
	[54]	No	Centralized	Date-driven (hybrid)
	Group/extend target	[22]	No	Centralized
[23]		Yes	Centralized	Data-driven (hybrid)
[24]		No	Centralized	Data-driven
[28]		Yes	Distributed	Model-driven

methods. There are also works studying using GP to represent the state-space model [49]. GP is used to learn the whole or part of the state-space model and the learned functions can be integrated into a particle filter or extended Kalman filter [50], [51], which results in hybrid tracking approach. Moreover, recently, different GP approaches are developed by assuming temporal and spatial correlation in the target trajectory and shape, and the state-space model is directly learned from the measurements [15], [16], [22]. In Table I, we have summarized and compared more existing relevant works in GP-assisted target tracking and localization. From Table I, we can find that although GP methods have been used for solving tracking and localization problems, there are rarely works about using GP in a distributed system and integrating data association solutions into GP methods.

The next section presents background knowledge for the GP methodology before presenting the distributed learning and tracking approaches.

III. BACKGROUND KNOWLEDGE

A. Standard Gaussian Process Method

GP is a stochastic process defining a distribution over functions that fit a set of points. Assume there exist correlations among target motions, at input $\mathbf{x} \in \mathbb{R}^d$, the non-linear mapping $f(\cdot)$ between the current input feature and the target state $f(\mathbf{x})$ can be modelled by a GP as

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (1)$$

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \quad (2)$$

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mathbf{x})(f(\mathbf{x}') - \mathbf{x}')], \quad (3)$$

where $m(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ denote the mean and covariance functions, respectively. The training and the test input data are denoted by \mathbf{x} and \mathbf{x}' , respectively. Popular examples of covariance functions include the squared exponential kernel. Namely,

$$k(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp \left\{ -\frac{1}{2} \sum_{j=1}^d \frac{(x_j - x'_j)^2}{l_j^2} \right\}, \quad (4)$$

where l_j represents the length-scale of the j^{th} feature of the input data. The length-scale describes how smooth a function is which can be thought of as roughly the distance you have to move in input space before the function value can change significantly [14]. The output variance σ^2 acts as a scaling factor. It determines the variation of function values from their mean.

The collected measurements can be treated as the noisy outputs of the unknown functions (which are target states in the tracking problem), therefore, a point target tracking problem with noisy observations can be written as

$$z = f(\mathbf{x}) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_z^2), \quad (5)$$

where z represents the measurement and ϵ represents the i.i.d. zero-mean Gaussian measurement noise with variance σ_z^2 .

Given a training data set of input-output pairs $D = \{\mathbf{X}, \mathbf{z}\}$ with $\mathbf{X} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_n^T]^T$ and $\mathbf{z} = [z_1, z_2, \dots, z_n]^T$, define $\mathbf{K} = k(\mathbf{X}, \mathbf{X})$ as the covariance matrix of the training input, and $\mathbf{k}_* = k(\mathbf{X}, \mathbf{x}_*)$ as the covariance between the training input \mathbf{X} and test input \mathbf{x}_* . To make a prediction of the target state at a new input \mathbf{x}_* , the predictive mean μ_* and the predictive variance σ_*^2 can be written as

$$\mu_* = m(\mathbf{x}_*) + \mathbf{k}_*^T \Sigma^{-1} (\mathbf{z} - m(\mathbf{x}_*)), \quad (6)$$

$$\sigma_*^2 = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T \Sigma^{-1} \mathbf{k}_*, \quad (7)$$

where $\Sigma = \mathbf{K} + \sigma_z^2 \mathbf{I}$, with \mathbf{I} being the identity matrix.

B. Distributed Gaussian Process Method

The computational complexity and storage cost are major challenges for the large-scale learning problems. The computations require $\mathcal{O}(N^3)$ time with a standard GP implementation, where N represents the number of the training instances. Besides, the standard GP also requires $\mathcal{O}(N^2 + Nd)$ of memory, where d is the dimensionality of the data. Both facts limit the scalability of the standard GP regression. Moreover, according to (6) and (7), the standard GP can only make predictions based on all the available data, which is a centralized scheme and requires data to be shared among local GPs.

In this section, inspired by the idea of divide-and-conquer, DGP methods are introduced to reduce not only the compu-

tational cost but also the memory cost of the standard GP, by first training local GPs based on subsets of the whole training data set and then aggregating the knowledge of local GPs to achieve more accurate high-level predictions. The overall computational complexity and the memory cost can be reduced to $\mathcal{O}(Nn^2)$ and $\mathcal{O}(Mn^2 + Nd)$ ($n \ll N$), respectively, where M represents the number of local GPs and n represents the size of data used for training a local GP. Particularly, both the computational complexity and storage cost can be further reduced through parallel/distributed computing [55].

The first type of DGP methods is the product of experts (PoEs) [37] approach. The idea is to multiply the local predictive probability distributions for overall predictions. Given the data $D^{(i)}$ collected by sensor i , the PoE predicts a function value $f(\mathbf{x}_*)$ at a corresponding test input \mathbf{x}_* according to

$$p(f(\mathbf{x}_*) | \mathbf{x}_*, D) = \prod_{i=1}^M p_i(f(\mathbf{x}_*) | \mathbf{x}_*, D^{(i)}), \quad (8)$$

where M is the number of GP experts and represents the number of active sensors which have measurements. Since the product of these Gaussian predictions is proportional to a Gaussian distribution, the aggregated predictive mean and variance can be calculated with closed form as

$$\mu_*^{\text{PoE}} = (\sigma_*^{\text{PoE}})^2 \sum_{i=1}^M \sigma_i^{-2}(\mathbf{x}_*) \mu_i(\mathbf{x}_*), \quad (9)$$

$$(\sigma_*^{\text{PoE}})^{-2} = \sum_{i=1}^M \sigma_i^{-2}(\mathbf{x}_*), \quad (10)$$

where $\mu_i(\mathbf{x}_*)$ and $\sigma_i^2(\mathbf{x}_*)$ represent the predictive mean and variance of GP expert i , respectively, which can be calculated based on (6) and (7).

The PoE model provides a straightforward way to aggregate local predictions and sidesteps the weight assignment issue in other DGP models such as the mixture of expert model [56]. However, this model becomes overconfident when making predictions, especially in regions without any training data.

The generalized product of experts (GPOEs) model [38] improves PoE by adding weights that represent the contributions of different experts. For instance, the weight can be calculated as the difference in the differential entropy between the prior distribution $p(f(\mathbf{x}_*))$ and the posterior predictive distribution $p(f(\mathbf{x}_*) | \mathbf{x}_*, D)$, which can be written as

$$\beta_i = 0.5 (\log \sigma_{**}^2 - \log \sigma_i^2(\mathbf{x}_*)), \quad (11)$$

where σ_{**}^2 represents the variance of the prior distribution $p(f(\mathbf{x}_*))$ and $\sigma_i^2(\mathbf{x}_*)$ denote the predictive variance of GP expert i , which can be calculated based on (7).

Given the data $D^{(i)}$ collected by sensor i , the GPOE predicts a function value $f(\mathbf{x}_*)$ at a test input \mathbf{x}_* . The predictive distribution and the closed forms of the aggregated predictive

mean and variance can be written as

$$p(f(\mathbf{x}_*) | \mathbf{x}_*, D) = \prod_{i=1}^M p_i^{\beta_i}(f(\mathbf{x}_*) | \mathbf{x}_*, D^{(i)}), \quad (12)$$

$$\mu_*^{\text{GPOE}} = (\sigma_*^{\text{GPOE}})^2 \sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) \mu_i(\mathbf{x}_*), \quad (13)$$

$$(\sigma_*^{\text{GPOE}})^{-2} = \sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*). \quad (14)$$

Alternatively, the Bayesian committee machine (BCM) [39] proposes to aggregate the experts' predictions from another view by imposing a conditional independence assumption that $D^{(i)} \perp\!\!\!\perp D^{(j)} | f(\mathbf{x}_*)$ which in turn explicitly introduces a common prior $p(f(\mathbf{x}_*) | \mathbf{x}_*)$ for the experts. Therefore, the BCM's prediction distribution can be written as

$$p(f(\mathbf{x}_*) | \mathbf{x}_*, D) = \frac{\prod_{i=1}^M p_i(f(\mathbf{x}_*) | \mathbf{x}_*, D^{(i)})}{p^{M-1}(f(\mathbf{x}_*) | \mathbf{x}_*)}, \quad (15)$$

where the denominator reaches an $(M-1)$ -fold division by the prior, which plays the role of a correction term that helps to recover the GP prior when leaving regions of training data. The closed form of the aggregated predictive mean and variance of the BCM can be calculated as

$$\mu_*^{\text{BCM}} = (\sigma_*^{\text{BCM}})^2 \sum_{i=1}^M \sigma_i^{-2}(\mathbf{x}_*) \mu_i(\mathbf{x}_*), \quad (16)$$

$$(\sigma_*^{\text{BCM}})^{-2} = \sum_{i=1}^M \sigma_i^{-2}(\mathbf{x}_*) + (1-M)\sigma_{**}^{-2}. \quad (17)$$

In [40], a robust Bayesian committee machine (RBCM) is proposed which combines both the features of the GPOE and BCM models. The predictive distribution and aggregated predictive mean and variance of the RBCM can be written as

$$p(f(\mathbf{x}_*) | \mathbf{x}_*, D) = \frac{\prod_{i=1}^M p_i^{\beta_i}(f(\mathbf{x}_*) | \mathbf{x}_*, D^{(i)})}{p^{\sum_i \beta_i - 1}(f(\mathbf{x}_*) | \mathbf{x}_*)}, \quad (18)$$

$$\mu_*^{\text{RBCM}} = (\sigma_*^{\text{RBCM}})^2 \sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) \mu_i(\mathbf{x}_*), \quad (19)$$

$$(\sigma_*^{\text{RBCM}})^{-2} = \sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) + (1 - \sum_{i=1}^M \beta_i) \sigma_{**}^{-2}. \quad (20)$$

All the models discussed in this section can be applied to infer the target states in a distributed way in the target tracking problem. Particularly, the closed-form of posterior predictions can be obtained and the predictions are fully tractable.

C. Hyperparameter Learning

The hyperparameters of GP need to be learned from the data. As a standard GP, maximum likelihood estimation (MLE) is applied to learn the hyperparameters by maximising the log marginal likelihood which can be written as

$$\log p(\mathbf{z} | \mathbf{X}, \boldsymbol{\theta}) = -\frac{1}{2} \mathbf{z}^\top \boldsymbol{\Sigma}^{-1} \mathbf{z} - \frac{1}{2} \log |\boldsymbol{\Sigma}| - \frac{n}{2} \log 2\pi, \quad (21)$$

where $\boldsymbol{\theta} = \{\sigma^2, \sigma_z^2, l_1, \dots, l_d\}$ represents the set of hyperparameters. l_j represents the length-scale of the j^{th} feature of the input data, σ^2 is the output variance of the kernel function, and σ_z^2 is the variance of the measurement noise.

For the DGP, assuming the local GPs are independent with each other, the log marginal likelihood can be factorized as

$$\begin{aligned} & \log p(\mathbf{z}|\mathbf{X}, \boldsymbol{\theta}) \\ & \approx \sum_{i=1}^M \log p_i(\mathbf{z}^{(i)}|\mathbf{X}^{(i)}, \boldsymbol{\theta}), \\ & = \sum_{i=1}^M \left(-\frac{1}{2} \mathbf{z}^{(i)\top} \boldsymbol{\Sigma}^{(i)-1} \mathbf{z}^{(i)} - \frac{1}{2} \log |\boldsymbol{\Sigma}^{(i)}| - \frac{n}{2} \log 2\pi \right), \end{aligned} \quad (22)$$

where $\mathbf{X}^{(i)}$ and $\mathbf{z}^{(i)}$ represent the training input and output of local GP i , respectively.

The reduction in computational complexity of DGP can also be justified by looking into (22), where both the computations of determinant and inversion are only based on a much smaller matrix $\boldsymbol{\Sigma}^{(i)}$. In addition, as compared to (21), the factorized marginal likelihood can potentially be maximized in a decentralized manner like federated learning [57] since it is a summation over local marginal likelihood functions.

The learned hyperparameters are shared by all the local experts for automatic regularization to avoid overfitting.

IV. DGP-BASED POINT TARGET TRACKING

The previous section demonstrates that the DGP is a promising method for large-scale learning systems. In this section, we describe the proposed DGPT approach which solves the tracking problem in WSNs using the distributed machine learning method, in a data-driven way. Several improvement schemes are designed to help integrate DGP for efficient distributed tracking and deal with clutter measurements to achieve robust performance.

In a WSN, each sensor can collect its own measurements and the GP regression can be applied locally to process local data. Some sensors can be edge devices and could provide edge learning [32]. The proposed DGPT is linked with federated learning [58].

At each time, after training, local GP-based target state estimations can be aggregated to reach a high-level prediction following different aggregation methods discussed in Section III-B. Only the predictive means and variances of local GPs are propagated to calculate the overall prediction, without transmitting all the data to a central filter (controller). Particularly, having this aggregation process does not mean an extra central node is necessary. The aggregation can be implemented on any capable sensor or edge node, thus the proposed approach is fully distributed.

A. Temporal and Spatial-Temporal GP

To make a state estimation of a target, it is reasonable to assume that there is a temporal correlation in the motions of the target, and this correlation within time variables in distant past is weaker than in more recent ones. Therefore, the target state can be formulated as a function of the time variable which is used as the input data for training DGP and making state estimations. In this case, we have $\mathbf{x} = t$, where t represents the

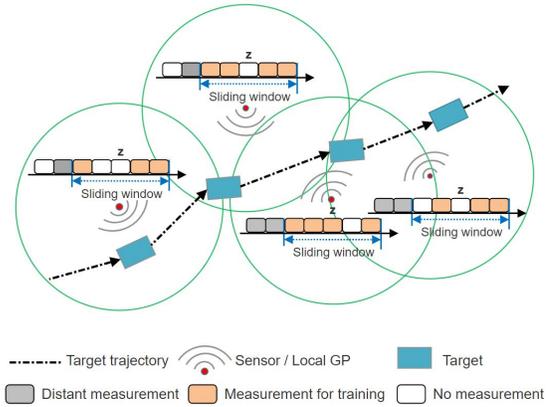


Figure 1: A distributed point tracking system with 4 sensors. The length of sliding window in this example is 5 time steps

time variable. Based on the temporal GP (TGP), the tracking problem in (1) and (5) can be reformulated as

$$f(t) \sim \mathcal{GP}(m(t), k(t, t')), \quad (23)$$

$$z = f(t) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_z^2), \quad (24)$$

where t and t' are the training and test data, respectively.

Based on the time variables and measurements (training set), set the test input as $\mathbf{x}_* = t$, TGP can predict the target state at time t using (6) and (7). Moreover, TGP can also predict a next-step state predictions by setting $\mathbf{x}_* = t + 1$.

Inspired by [59] that uses GP regression to learn the target state transition function, the spatial correlation can also be involved for state prediction by including the target state of the previous time into the input data, namely $\mathbf{x} = (r_{t-1}, t)$, where r_{t-1} represents the target state in $t - 1$. The spatial-temporal GP (STGP)-based tracking problem can be formulated as

$$f(r_{t-1}, t) \sim \mathcal{GP}(m(r_{t-1}, t), k(r_{t-1}, t; r'_{t-1}, t')), \quad (25)$$

$$z = f(r_{t-1}, t) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_z^2). \quad (26)$$

B. Sliding Window-based Tracking

Although DGP is designed for complexity reduction, training GPs distributively can still be computationally intense when local sensors collect extensive measurements originating from both the targets and the clutter.

Noticing the fact that the target motion can be a mixed maneuvering behaviour with time-varying parameters, based on the temporal correlation assumptions in the previous section, measurements with weaker motion correlations in distant past time cannot contribute to training and prediction as much as the recent ones. Therefore to further reduce the computational complexity, the distant measurements are abandoned in DGPT and only the recent measurements are utilized for training, which can be treated as using a sliding window to select valid measurements. The sliding window is a set of time variables which represents a number of time steps. Based on the sliding window, a sensor without valid measurements (the times of collecting the measurements are not in the sliding window) is excluded from state estimation at the current time.

When both the number of experts and target measurements is reduced, the computational complexity of GP training reduces and also the tracking accuracy improves. Figure 1 shows the framework of the proposed sliding window-based DGPT approach.

C. DGP-based Data Association

The previous section discuss using sliding window to reduce the number of measurements for training and state estimation. In this section, we focus on dealing with a large number of clutter measurements to improve the tracking accuracy and also reduce the computational complexity.

To cope with the clutter and for learning the GP hyperparameters, a method is designed to assign weights to different measurements based on the marginal likelihood (21). A weighted summation over measurements collected by one sensor is then calculated as the training data for DGP. Define $j \in \{1, 2, \dots, J_t\}$ as the index of measurements received by a sensor at time t and define $w_{j,t}$ as the weight of the j^{th} measurement, the gating method can be written as

$$w_{j,t} = p(z_j | \mathbf{X}_t, \boldsymbol{\theta}_{t-1}) / \sum_j^{J_t} p(z_j | \mathbf{X}_t, \boldsymbol{\theta}_{t-1}), \quad (27)$$

$$\bar{z}_t = \sum_j^{J_t} w_{j,t} z_j, \quad (28)$$

where \bar{z}_t represents the weighted summation of the measurements, which will be used for hyperparameter learning. The rationale for calculating this weighted summation is to determine a training instance based on the likelihoods of all the measurements (both target and clutter measurements) and the resulting summation is expected to be close to the target measurements. Particularly, since only one summation is calculated from each local expert, the number of measurements for DGP training is greatly reduced and therefore the computational cost is reduced as well.

In the distributed tracking scenario, at the beginning of each time, each sensor performs the proposed gating method independently based on the learned hyperparameters from the previous time and the local measurements which are inside the sliding window. The resulting data based on (27) and (28) is used for hyperparameter learning and state estimation in the current time. In addition, for a new sensor that does not have any historical measurements, average weights are calculated according to the weights assigned by all the other active sensors. Moreover, when the clutter rate is much higher, the clustering scheme can be used for preprocessing and the gating method will be applied to the cluster centers rather than to all the measurements.

D. Hyperparameter Online Learning in DGPT

Since the target motion can be time-varying and the sliding window is designed to keep valid measurements for training and tracking, the hyperparameters of DGP should be learned online to capture the non-stationary features. Hence, in the proposed DGPT approach, MLE which is based on the factorized marginal likelihood (22) is solved every time to update the hyperparameters. This process brings extra computational

costs due to the non-convexity of (22) and requires an iterative solving process. To accelerate the hyperparameter learning process, optimized hyperparameters at time t is designed to be set as the initial value of hyperparameters of MLE at $t+1$, which can significantly reduce the iterations needed for MLE.

Benefiting from the proposed sliding window design, the DGP-based data association, and the online learning properties, the DGPT can provide accurate target state estimations with collected measurements.

E. Complexity Analysis

For the proposed DGPT approach, the main computational complexity stems from the covariance matrix inversion, which scales cubically in terms of the number of data at each sensor. Based on [23], the complexity of matrix inversion for DGP is $\mathcal{O}(J_{1,t}^3 + J_{2,t}^3 + \dots + J_{M_t,t}^3)$, where $J_{M_t,t}$ represents the number of data stored in the M_t^{th} active sensor at time t and M_t is the number of active sensors at t . Based on (27) and (28), the measurements collected by a sensor at time t are used to calculate a weighted sum for DGP training. This means only a single measurement is saved per sensor per time. Given the length of the sliding window of time as C , the computational complexity due to the matrix inversion can be upper bounded as $\mathcal{O}(C^3 M_t)$.

In the DGPT approach, the hyperparameters are learned online by solving the maximum likelihood estimation problem as formulated in (22). Maximizing this function requires computing the covariance matrix inversion iteratively. Therefore, the computational complexity of the DGP algorithm update scales as $\mathcal{O}(C^3 M_t)$ per iteration at time t . The prediction step also scales as $\mathcal{O}(C^3 M_t)$. It is important to notice that the proposed DGP-based data association helps the DGPT achieve a low computational complexity in both DGP learning and state prediction since the complexity does not scale with the number of received measurements and the length of the sliding window does not increase over time.

V. THEORETICAL PERFORMANCE ANALYSIS

Lemma 1. (Lemma 5.1 of [60]) *Given a trained local GP based on training data $D_t = \{\mathbf{X}_t, \mathbf{z}_t\}$ till t , for any input $\mathbf{x}_* \in \mathbf{X}_t$, the probability $\Pr(\cdot)$ that the predictive mean $\mu(\mathbf{x}_*)$ deviates from the true function value by more than a certain amount can be upper bounded as*

$$\Pr \left\{ |f(\mathbf{x}_*) - \mu(\mathbf{x}_*)| > \gamma^{1/2} \sigma(\mathbf{x}_*) \right\} \leq e^{-\gamma/2}, \quad (29)$$

where γ is a positive constant.

Lemma 1 proposes a UCB of the probability that the deviation between the true function and the estimated mean of the function is larger than a scaled version of the estimated variance function. Based on this lemma, error bounds of distributed GPs can be derived.

The following Lemma 2 represents a generalization of Lemma 1, assume each time one estimation is made, for the case of an infinite number of time t , e.g. $t \rightarrow \infty$.

Lemma 2. Define $\delta \in (0, 1)$, set $\gamma_t = 2 \log(\pi_t/\delta)$, for $\pi_t = \pi^2 t^2/6$, based on Lemma 1, apply the union bound over t , we have

$$\Pr \left\{ \bigcup_{t=1}^{\infty} |f(\mathbf{x}_t) - \mu(\mathbf{x}_t)| > \gamma_t^{1/2} \sigma(\mathbf{x}_t) \right\} \leq \sum_{t=1}^{\infty} e^{-\gamma_t/2} = \sum_{t=1}^{\infty} \frac{\delta}{\pi_t} = \delta. \quad (30)$$

Lemma 2 further generalises the cumulative deviation of the predictive mean from the true function value based on the GP predictions from all n test inputs.

Theorem 1. (One-step error bound of GPoE) Consider a distributed GP system with M local GPs, with probability at least $1 - \sum_{i=1}^M e^{-\gamma_i/2}$, the deviation between the true function value at \mathbf{x}_* and the aggregated estimation of the mean value made by the GPoE method can be upper bounded as

$$|f(\mathbf{x}_*) - \mu_*^{\text{GPoE}}| \leq \frac{\sum_{i=1}^M \gamma_i^{1/2} \sigma_i^{-1}(\mathbf{x}_*)}{\sum_{i=1}^M \sigma_i^{-2}(\mathbf{x}_*)}. \quad (31)$$

Proof: Define A_i as the event in which the prediction of the target state from local expert i and the true target state differs larger than a quantity, which can be written as

$$A_i = \left\{ |f(\mathbf{x}_*) - \mu_i(\mathbf{x}_*)| > \gamma_i^{1/2} \sigma_i(\mathbf{x}_*) \right\}. \quad (32)$$

Define the union of events $\{A_1, A_2, \dots, A_M\}$ as A . Applying the union bounds over M events, the probability of A can be upper bounded as

$$\begin{aligned} \Pr(A) &:= \Pr \left\{ \bigcup_{i=1}^M A_i \right\}, \\ &= \Pr \left\{ \bigcup_{i=1}^M |f(\mathbf{x}_*) - \mu_i(\mathbf{x}_*)| > \gamma_i^{1/2} \sigma_i(\mathbf{x}_*) \right\}, \\ &\leq \sum_{i=1}^M e^{-\gamma_i/2}, \end{aligned} \quad (33)$$

where $\mu_i(\mathbf{x}_*)$ and $\sigma_i(\mathbf{x}_*)$ represent the predictive mean and standard deviation (STD) of local GP i at \mathbf{x}_* , respectively.

Define \bar{A} as the complement of A , changing the direction of the inequality gives that

$$\begin{aligned} \Pr(\bar{A}) &= \Pr \left\{ \bigcap_{i=1}^M \bar{A}_i \right\}, \\ &= \Pr \left\{ \bigcap_{i=1}^M |f(\mathbf{x}_*) - \mu_i(\mathbf{x}_*)| \leq \gamma_i^{1/2} \sigma_i(\mathbf{x}_*) \right\}, \\ &\geq 1 - \sum_{i=1}^M e^{-\gamma_i/2}. \end{aligned} \quad (34)$$

According to (13) and (14), the deviation between the true function value and the aggregated predictive mean by GPoE can be written as

$$\begin{aligned} \left| f(\mathbf{x}_*) - \mu_*^{\text{GPoE}} \right| &= \left| f(\mathbf{x}_*) - \frac{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) \mu_i(\mathbf{x}_*)}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*)} \right|, \\ &= \frac{|\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) (f(\mathbf{x}_*) - \mu_i(\mathbf{x}_*))|}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*)}, \\ &\leq \frac{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) |f(\mathbf{x}_*) - \mu_i(\mathbf{x}_*)|}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*)}. \end{aligned} \quad (35)$$

Based on (34), with probability at least $1 - \sum_{i=1}^M e^{-\gamma_i/2}$, (35) can be upper bounded as

$$\begin{aligned} |f(\mathbf{x}_*) - \mu_*^{\text{GPoE}}| &\leq \frac{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) \gamma_i^{1/2} \sigma_i(\mathbf{x}_*)}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*)}, \\ &= \frac{\sum_{i=1}^M \beta_i \gamma_i^{1/2} \sigma_i^{-1}(\mathbf{x}_*)}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*)}, \end{aligned} \quad (36)$$

which completes the proof.

Theorem 1 proposes a theoretical UCB for the tracking performance. Define the highest predictive variances of a local expert as σ_H^2 , the bound can be further represented as

$$\begin{aligned} 1 - \sum_{i=1}^M e^{-\gamma_i/2} &\leq \Pr \left\{ |f(\mathbf{x}_*) - \mu_*^{\text{GPoE}}| \leq \frac{\sum_{i=1}^M \beta_i \gamma_i^{1/2} \sigma_i^{-1}(\mathbf{x}_*)}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*)} \right\}, \\ &\leq \Pr \left\{ |f(\mathbf{x}_*) - \mu_*^{\text{GPoE}}| \leq \frac{\sum_{i=1}^M \beta_i \gamma_i^{1/2} \sigma_i^{-1}(\mathbf{x}_*)}{M \sigma_H^{-2}(\mathbf{x}_*) \sum_{i=1}^M \beta_i} \right\}. \end{aligned}$$

This bound demonstrates that, given all other local GPs fixed, when one of the local GP makes a highly uncertain prediction which is reflected as a larger predictive variance, the upper bound of the deviation will increase, which means the overall prediction is exacerbated by this poor GP expert.

Next, we derive a Theorem about the UCB of the RBCM.

Theorem 2. (One-step error bound RBCM) following Theorem 1, for a distributed GP system with M local GPs, with probability at least $1 - \sum_{i=1}^M e^{-\gamma_i/2}$, the deviation between the true function value at \mathbf{x}_* and the aggregated estimation of the mean value made by the RBCM method can be upper bounded as

$$|f(\mathbf{x}_*) - \mu_*^{\text{RBCM}}| \leq \frac{\sum_{i=1}^M \gamma_i^{1/2} \sigma_i^{-1}(\mathbf{x}_*)}{\sum_{i=1}^M \sigma_i^{-2}(\mathbf{x}_*)}. \quad (37)$$

Proof: The detailed proof is given in Appendix A.

The UCB for the GPoE algorithm in general is not the same as the UCB (31) for the RBCM (37). However, under certain conditions the two upper bounds coincide.

The next Theorem 3 generalises the result further, to the GPoE's cumulative error bound for a number of estimations over a wide time interval T . The cumulative error bound of other DGPT approaches can be derived in a similar way.

Theorem 3. (Cumulative error bound) Consider a distributed GP system with M local GPs. suppose one state estimation is made every time, with probability at least $1 - \sum_{i=1}^M \delta_i$, the cumulative deviation between the true function value at each test input and the aggregated estimation of the mean value can be upper bounded as

$$\sum_{t=1}^T |f(\mathbf{x}_t) - \mu^{\text{GPoE}}(\mathbf{x}_t)| \leq \sum_{t=1}^T \frac{\sum_{i=1}^M \beta_i \gamma_{t,i}^{1/2} \sigma_i^{-1}(\mathbf{x}_t)}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_t)}. \quad (38)$$

Proof: The detailed proof is given in Appendix B.

VI. DGP-ASSISTED BAYESIAN FILTERING WITH MEASUREMENT ORIGIN UNCERTAINTY

Inspired by ideas from [61]–[63], in this section, the proposed DGPT is enhanced by an elegant Bayesian filtering method which can solve the data association problem without the need to construct explicit measurement-target assignment hypotheses or gates. The resultant hybrid Bayesian filtering-based tracking approach provides a novel way to merge distributed machine learning and model-based Bayesian inference. The prediction made from the DGPT is used as the prior distribution of the target state and a Poisson measurement likelihood model is involved for posterior state inference.

A. Measurement Likelihood Function

According to [61], the state vector of $L + 1$ entities that needs to be estimated at time t is defined as

$$\tilde{\mathbf{X}}_t = [\tilde{\mathbf{x}}_{t,c}^\top, \tilde{\mathbf{x}}_{t,1}^\top, \dots, \tilde{\mathbf{x}}_{t,L}^\top]^\top, \quad (39)$$

where $\tilde{\mathbf{x}}_{t,c}$ represents the state of the clutter process and L represents the number of targets which is assumed to be known. In a single point tracking problem, we have $L = 1$ and $\tilde{\mathbf{x}}_{t,1} = \tilde{\mathbf{x}}_t$.

The derivation of the measurement likelihood relies on the following three assumptions:

- A1:** The numbers of target originated measurements in a time scan are assumed to be Poisson distributed, with a rate λ_T .
- A2:** The numbers of clutter measurements in a time scan are assumed to be Poisson distributed, with a clutter rate λ_c .
- A3:** The clutter measurements are assumed to be uniformly distributed in the sensing space of each sensor.

In many cases, in a time step, a high-resolution sensor is able to generate more than one measurement from the target and also from the environmental interference. Therefore, the assumptions that the clutter rate λ_c and target rate λ_T of the respected measurements follow Poisson distributions are well justified. These assumptions are also used in [64], [65]. Then this is reflected in the first and second assumptions (A1 and A2). Assumption A3 reflects the fact that the clutter is uniformly distributed which is one of the most common cases in practice. The uniform distribution of the clutter in the areas of interest also reflects the full lack of prior knowledge about the possible locations of the environmental interference.

For a WSN, according to above assumptions and by assuming that the target originated measurement likelihood is a product of Gaussian likelihoods, the collected measurements result from superposition of multiple target measurements with Gaussian noise and uniform clutter measurements [61]. Therefore, similarly to [62] and [63], the joint likelihood expression of the tracking problem can be written as

$$p(\mathbf{z}_t | \tilde{\mathbf{X}}_t) = p((z_{t,1}, z_{t,2}, \dots, z_{t,n_t}), n_t | \tilde{\mathbf{X}}_t), \quad (40)$$

where n_t represents the number of measurements collected from all the sensors in time t . \mathbf{z}_t denotes the measurements collected in t .

To simplify the notation we rewrite $\tilde{\mathbf{x}}_t$ as \tilde{x}_t to represent the case when the target state is a scalar.

Define $p_c(z_{t,j})$ as the clutter measurement likelihood which is a uniform distribution, and define $p(z_{t,j} | \tilde{x}_t)$ as the target measurement likelihood which is Gaussian, based on the assumptions. In addition, define ϕ as a partition of the measurement set, the joint likelihood (40) can be written as

$$p(\mathbf{z}_t | \tilde{\mathbf{X}}_t) \propto \sum_{\phi} \left(\frac{\lambda_T}{\lambda_c} \right)^{n_t^\top(\phi)} \prod_{j, \phi(j) \neq 0}^{n_t} p(z_{t,j} | \tilde{x}_t) \prod_{j, \phi(j) = 0}^{n_t} p_c(z_{t,j}), \quad (41)$$

where $n_t^\top(\phi)$ represents the number of target measurements which is compatible with partition ϕ . $\phi(j) = 0$ corresponds to the clutter measurement and $\phi(j) \neq 0$ corresponds to the target measurement. λ_T is the expected number of measurements originating from the target and λ_c denotes the expected number of measurements corresponding to clutter in the sensing area. According to Appendix A of [63], the likelihood can be further represented as

$$p(\mathbf{z}_t | \tilde{\mathbf{X}}_t) \propto \prod_{j=1}^{n_t} (\lambda_c p_c(z_{t,j}) + \lambda_T p(z_{t,j} | \tilde{x}_t)). \quad (42)$$

Based on Assumption A3, the clutter measurement likelihood can be written as $p_c(z_{t,j}) = \frac{1}{A_{\text{sen}}}$, where A_{sen} represents the sensing area and is considered to be the same for all the sensors.

The measurement likelihood is expressed in two different ways: with the Poisson likelihood model (42) or its equivalent form (43) and with a Gaussian process as in (45).

Assuming the measurement noise follows a zero-mean Gaussian distribution with variance σ_z^2 , we have $p(z_{t,j} | \tilde{x}_t) \sim \mathcal{N}(\tilde{x}_t, \sigma_z^2)$, the measurement likelihood (42) can be represented as

$$p(\mathbf{z}_t | \tilde{\mathbf{X}}_t) \propto \prod_{j=1}^{n_t} \left\{ \frac{\lambda_c}{A_{\text{sen}}} + \lambda_T p(z_{t,j} | \tilde{x}_t) \right\}, \quad (43)$$

$$\propto \prod_{j=1}^{n_t} \mathcal{N}(\hat{\mu}_{t,j}, \hat{\sigma}_{t,j}^2), \quad (44)$$

$$\propto \mathcal{N}(\hat{\mu}_t, \hat{\sigma}_t^2), \quad (45)$$

where $\hat{\mu}_{t,j} = \frac{\lambda_c}{A_{\text{sen}}} + \lambda_T \tilde{x}_t$ and $\hat{\sigma}_{t,j}^2 = (\lambda_T \sigma_z)^2$. The derivation from (44) to (45) holds due to the fact that the product of Gaussian probability density functions is proportional to a Gaussian probability density function.

Based on (9) and (10) of product of Gaussian, the mean and the variance of (45) can be calculated as

$$\hat{\sigma}_t^2 = \frac{1}{\sum_{j=1}^{n_t} \hat{\sigma}_{t,j}^{-2}} = \frac{1}{\sum_{j=1}^{n_t} (\lambda_T \sigma_z)^{-2}} = \frac{(\lambda_T \sigma_z)^2}{n_t}, \quad (46)$$

$$\begin{aligned} \hat{\mu}_t &= \hat{\sigma}_t^2 \sum_{j=1}^{n_t} \hat{\sigma}_{t,j}^{-2} \hat{\mu}_{t,j} = \hat{\sigma}_t^2 \sum_{j=1}^{n_t} (\lambda_T \sigma_z)^{-2} \cdot \left(\frac{\lambda_c}{A_{\text{sen}}} + \lambda_T \tilde{x}_t \right), \\ &= \hat{\sigma}_t^2 n_t (\lambda_T \sigma_z)^{-2} \cdot \left(\frac{\lambda_c}{A_{\text{sen}}} + \lambda_T \tilde{x}_t \right), \\ &= \frac{\lambda_c}{A_{\text{sen}}} + \lambda_T \tilde{x}_t. \end{aligned} \quad (47)$$

The expressions (46) and (47) are functions of the clutter parameter (e.g. of the clutter rate λ_c) and of the rate λ_T of the

target originated measurement. The likelihood function could be expressed also as a function of the clutter density, as this is shown in [62], [63]. The clutter rate λ_c and the clutter density are connected with the area of the sensor and are parameters that can be estimated by knowing the sensor area A_{sen} , as shown in [66] or by theoretical methods.

B. DGP-based Posterior State Inference

As discussed in Section IV-A, notice that at time t , the proposed DGPT approach can also provide a next-step state estimation distribution. This means the DGP model can be used as a high-quality prior distribution for the state estimation in the next time. Have this prior knowledge and by combining the likelihood, a novel state estimation method can be designed.

Define $\mathbf{Z}_{t-C:t-1}$ as the set of measurements within the sliding window from time $t - C$ to $t - 1$. The length of the sliding window is C time steps. Given the current time t , the prior distribution of the target state can be written as

$$p(\tilde{\mathbf{X}}_t | \mathbf{Z}_{t-C:t-1}) \sim \mathcal{N}(\mu_{t-1}, \sigma_{t-1}^2), \quad (48)$$

where μ_{t-1} and σ_{t-1}^2 represent the prior mean and variance that can be calculated using the GP regression equations (4) and (5), respectively. According to Bayes rule, the posterior can be written as

$$p(\tilde{\mathbf{X}}_t | \mathbf{Z}_{t-C:t}) \propto p(\mathbf{z}_t | \tilde{\mathbf{X}}_t) p(\tilde{\mathbf{X}}_t | \mathbf{Z}_{t-C:t-1}). \quad (49)$$

Based on the prior distribution (18), (19), and (20), which is learned by the DGP, considering the measurement likelihood (42), the posterior state distribution (49) can be derived. The posterior mean and variance can be written as

$$\mu_t = \frac{\mu_{t-1} \hat{\sigma}_t^2 + \sigma_{t-1}^2 \sum_{j=1}^{n_t} (z_j \lambda_T - \lambda_T \lambda_c / A_{\text{sen}})}{\hat{\sigma}_t^2 + n_t \sigma_{t-1}^2 \lambda_T^2}, \quad (50)$$

$$\sigma_t^2 = \frac{\hat{\sigma}_t^2 \sigma_{t-1}^2}{\hat{\sigma}_t^2 + n_t \sigma_{t-1}^2 \lambda_T^2}. \quad (51)$$

Appendix C contains the detailed derivation of these results.

A thorough performance validation and evaluation of the developed DGPT and hybrid Bayesian filtering approaches with the derived theoretical UCB and over several test scenarios are presented in the next section.

VII. PERFORMANCE EVALUATION AND VALIDATION

A. Training Time of DGP

To evaluate the training time of the DGP model, in this subsection, we define the input data as two-dimensional, namely $\mathbf{x} = [x_1, x_2]^T$, we test the running time of DGP training based on data from the following function

$$f(\mathbf{x}) = 5x_1^2 + \sin(12x_2), \quad (52)$$

where the measurements are generated by adding zero-mean Gaussian noise with the STD as $\sigma_z = 0.5$ to this function.

The time required to compute the log marginal likelihood and its gradient with respect to the kernel hyperparameters is presented in Figure 2. This computation process acts as the fundamental step for solving the factorized MLE (22) for hyperparameter learning and DGP training. As a comparison,

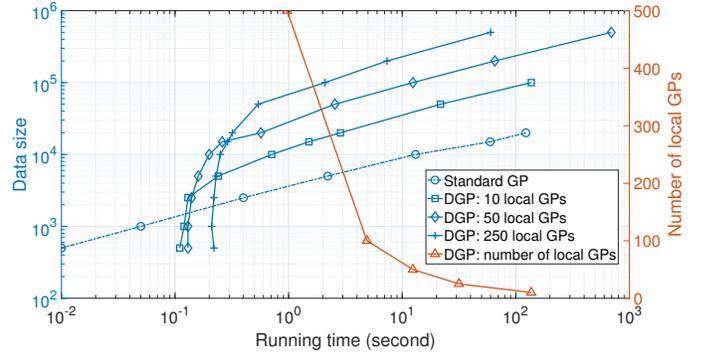


Figure 2: Computation time for the log-marginal likelihood and its gradient versus the size of the training data and the number of local GPs

we also measure the time required for training the standard GP model, which corresponds to solving the original MLE (21). In addition, we measure the running time of solving the factorized MLE with varying numbers of local GPs. According to the results, we can find that the computation time of training DGP increases with the growing size of the training data set and adding more local GPs can accelerate the training process.

Particularly, except in the cases of small training sets (500-1000), the running time of DGP increases much slower than the standard GP. Hence, DGP can handle much more data and is more suitable for real-time distributed learning and tracking in WSNs as compared to the standard GP. Moreover, the computation of the factorized likelihood can be implemented in parallel, thus each local computational unit will only need to compute one or a few terms of (22), and the overall running time can be further reduced.

In addition, in Figure 2, the impact of the number of local GPs on the running time for computing the log marginal likelihood and its gradient is also presented based on the data set with a fixed size of 10000. We can find that having more local GPs can help reduce the running time since according to (21), if the smaller size of covariance matrices are generated, solving the matrix determinant and inversion can be much faster on these smaller matrices.

B. Simulation Setup

The tracking performance of the proposed approaches are tested in a WSN with 250 sensors uniformly implemented in a 1000 meters \times 1000 meters area. The sensing range is 50 meters and the sampling period is one second, both of which are identical for every sensor. The proposed algorithm can also be implemented in a heterogeneous network easily and can make state estimations considering the heterogeneity of sensors, which for example, can be reflected in the posterior predictive variance of the local sensor.

In this paper, the target states denotes the target locations, hence two GPs are needed at one sensor for the X -coordinate and Y -coordinate, respectively. For the GP, a zero-mean function is used which means no extra knowledge is utilized for tracking. Besides, the covariance function is selected to be the squared exponential kernel which is demonstrated to

perform well in many maneuvering models [17]. Following Assumptions A1-A3, the clutter measurements are modelled as a Poisson point process and are uniformly located in the sensing region of each active sensor. Moreover, the number of target measurements is modelled as a zero-truncated Poisson distribution, which ensures at least one measurement can be received by a sensor within a time step, namely, $\lambda_T = 1$. The target measurement noise is modelled by the zero-mean Gaussian distribution.

In addition, we develop a range of scenarios with varying parameters of the clutter process, measurement noise, and target trajectories. To test the robustness of the proposed DGPT approach and also to account for the wireless effect of the channel, three noise levels are involved in the simulation with measurement uncertainty (STD) $\sigma_z = 1, 2, 4$ meters. In addition, two clutter settings are simulated to test the performance of the proposed approach, the low clutter case sets the clutter rate as 1 and the high clutter case sets the rate to be 5. All the results are averaged over 100 Monte Carlo (MC) runs and the lengths of sliding windows in different trajectories are carefully tuned to be different for optimal performance.

C. Benchmarks

Since this paper focuses on the model-free approaches, the standard GP-based centralized tracking approach is simulated as the benchmark. This scheme relies on solving the MLE (21) to learn the hyperparameters, and the learning process requires the measurements to be transmitted in the WSN. To make fair comparisons, the standard GP-based centralized tracking approach is trained with the same sensor measurements in the sliding window, which means this approach uses the same amount of data for model training and hyperparameter learning as well as the DGPT approach.

To study the impact of different aggregation methods on the DGPT, both RBCM and GPoE are simulated. In addition, DGPTs based on both temporal and spatial-temporal input data are evaluated. For the temporal case, a set of time variables which is in the sliding window is used as the input data. For the spatial-temporal case, both the current time and the target state of the previous time are used as input. Notice that in the online tracking problem, the real target state is not available to be used as the training input. Therefore, the predictive state acquired by the DGPT in the previous time is used instead.

D. Target Trajectories

To evaluate the proposed algorithm and the benchmarks, four challenging scenarios are built following different models. The trajectories and the sensors are depicted in Fig. 3.

- S1:** Similar to [67], the trajectory is generated based on the NCV model in the straight line, and the abrupt velocity change at each pre-defined turning point.
- S2:** The target trajectory is generated by the gradual coordinated turns model ($20^\circ/s$ for 10 seconds) with the constant velocity model, which can go both left and right. This model can represent maneuverable motions.

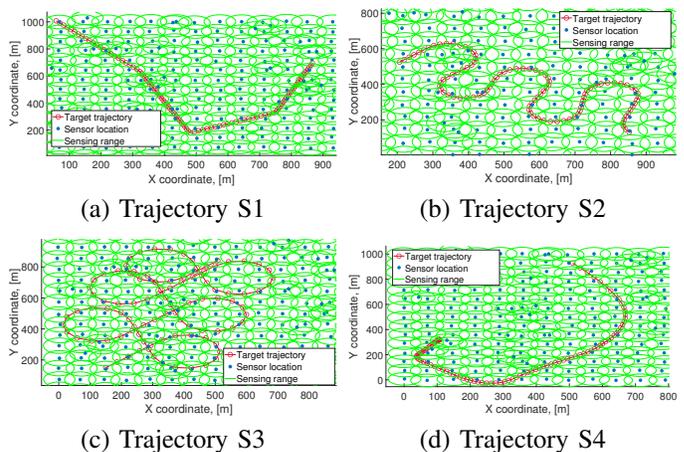


Figure 3: Target trajectories

- S3:** The target trajectory is generated by the sharp coordinated turns model which is more agile with higher turning rate ($30^\circ/s$ for 9 s).
- S4:** The target trajectory is generated by the Singer acceleration model. The maximum possible acceleration is $50 m/s^2$, the probability of non-acceleration is 0.4.

E. Normalized Root Mean Square Errors

In this section, the average normalized root mean square errors (NRMSEs) of the proposed DGPT and the standard GP-based tracker are evaluated. The NRMSE is defined as

$$\text{NRMSE} = \sqrt{\frac{\sum_i^N (\hat{f}(\mathbf{x}_i) - f(\mathbf{x}_i))^2}{N}} / (f(\mathbf{x}_i)_{\max} - f(\mathbf{x}_i)_{\min}),$$

where $f(\mathbf{x}_i)_{\max}$ and $f(\mathbf{x}_i)_{\min}$ represent the maximum and minimum value of the target states, respectively.

The tracking errors of the proposed DGPT approach with temporal input feature, which are collected from different trajectories are presented in Tables II-V. From the results, we can find that when considering the existence of clutter measurements, the clutter rate plays an important role in determining the prediction error. By contrast, the proposed DGPT approach achieves robust performance while changing the noise level of the target measurements. Moreover, using RBCM as the prediction aggregation method can achieve lower NRMSEs than GPoE in most scenarios, which justifies that adding the common prior into the aggregation process can improve the tracking accuracy. Particularly, the proposed DGPT approach performs competitively well and even outperforms the centralized approach in some scenarios. This is due to that in DGP, different weights can be assigned to the local predictions during the prediction aggregation process, so the final aggregated predictions are closer to the expert who makes more confident predictions. In the centralized method, all the data is aggregated before training without any difference.

The tracking errors of the proposed DGPT approach with both temporal and spatial input features are depicted in Tables

Table II: Updated NRMSEs for S1: TGP

Noise level	Clutter rate	Approach					
		Standard GP		DGPT-RBCM		DGPT-GPoE	
		X	Y	X	Y	X	Y
1	1	0.77%	1.15%	0.93%	0.95%	1.51%	1.20%
	5	2.12%	1.78%	2.10%	2.14%	2.91%	2.72%
2	1	0.79%	1.15%	0.94%	0.85%	1.60%	1.17%
	5	2.02%	1.94%	2.17%	2.57%	2.97%	3.11%
3	1	0.84%	1.19%	0.98%	0.93%	1.70%	1.34%
	5	2.05%	1.91%	2.25%	3.01%	2.98%	3.59%

Table III: Updated NRMSEs for S2: TGP

Noise level	Clutter rate	Approach					
		Standard GP		DGPT-RBCM		DGPT-GPoE	
		X	Y	X	Y	X	Y
1	1	1.22%	1.15%	1.50%	1.63%	2.04%	1.92%
	5	2.69%	2.75%	2.60%	3.00%	2.99%	3.27%
2	1	1.25%	1.19%	1.43%	1.55%	1.85%	1.78%
	5	2.77%	2.85%	2.62%	3.33%	2.94%	3.65%
3	1	1.32%	1.34%	1.55%	1.79%	1.87%	2.08%
	5	2.81%	2.90%	2.73%	3.50%	3.04%	3.87%

Table IV: Updated NRMSEs for S3: TGP

Noise level	Clutter rate	Approach					
		Standard GP		DGPT-RBCM		DGPT-GPoE	
		X	Y	X	Y	X	Y
1	1	1.82%	1.91%	1.81%	1.63%	1.82%	1.66%
	5	3.17%	3.40%	3.64%	3.23%	3.69%	3.29%
2	1	1.81%	1.95%	1.84%	1.65%	1.85%	1.68%
	5	3.19%	3.42%	3.64%	3.21%	3.69%	3.27%
3	1	1.86%	1.99%	1.88%	1.70%	1.90%	1.73%
	5	3.20%	3.43%	3.66%	3.23%	3.72%	3.29%

Table V: Updated NRMSEs for S4: TGP

Noise level	Clutter rate	Approach					
		Standard GP		DGPT-RBCM		DGPT-GPoE	
		X	Y	X	Y	X	Y
1	1	0.98%	0.70%	0.87%	0.56%	1.13%	0.73%
	5	1.64%	1.59%	2.77%	1.90%	3.88%	2.53%
2	1	0.99%	0.71%	1.04%	0.61%	1.37%	0.80%
	5	1.75%	1.48%	2.83%	1.92%	3.95%	2.58%
3	1	1.05%	0.73%	1.36%	0.79%	1.97%	1.10%
	5	1.87%	1.48%	3.02%	2.02%	4.21%	2.67%

Table VI: Updated NRMSEs for S1: STGP

Noise level	Clutter rate	Approach					
		Standard GP		DGPT-RBCM		DGPT-GPoE	
		X	Y	X	Y	X	Y
1	1	0.73%	0.80%	0.81%	0.82%	1.27%	1.02%
	5	1.93%	1.52%	2.64%	1.98%	3.31%	2.18%
2	1	0.73%	0.81%	0.84%	0.84%	1.35%	1.03%
	5	1.92%	1.59%	2.68%	2.07%	3.34%	2.25%
3	1	0.80%	0.88%	0.93%	0.91%	1.59%	1.09%
	5	1.95%	1.59%	2.64%	2.04%	3.24%	2.28%

VI-IX. We can find that considering spatial feature can help to improve the tracking accuracy as compared to TGP, especially in the more challenging scenarios where the speed of the target keeps changing or the target keeps maneuvering (Scenarios S2,s3, and S4). Particularly, the improvement is even more significant in the high clutter case since adding spatial feature can help to learn a more accurate likelihood function which can better assign weights for measurement preprocessing.

F. Uncertainty Quantification

To visualise the derived UCB of the proposed DGPT, we choose the probability that the UCB holds as 99.7% which

Table VII: Updated NRMSEs for S2: STGP

Noise level	Clutter rate	Approach					
		Standard GP		DGPT-RBCM		DGPT-GPoE	
		X	Y	X	Y	X	Y
1	1	1.07%	1.38%	1.29%	1.67%	2.48%	2.15%
	5	2.02%	2.21%	2.10%	2.44%	3.31%	2.86%
2	1	1.10%	1.40%	1.30%	1.72%	2.54%	2.13%
	5	2.01%	2.19%	2.14%	2.47%	3.41%	2.82%
3	1	1.15%	1.47%	1.35%	1.71%	2.62%	2.09%
	5	2.03%	2.18%	2.18%	2.46%	3.26%	2.82%

Table VIII: Updated NRMSEs for S3: STGP

Noise level	Clutter rate	Approach					
		Standard GP		DGPT-RBCM		DGPT-GPoE	
		X	Y	X	Y	X	Y
1	1	3.25%	2.93%	1.42%	1.24%	1.47%	1.21%
	5	2.91%	4.09%	2.41%	1.85%	2.49%	1.86%
2	1	3.43%	2.93%	1.66%	1.19%	1.66%	1.23%
	5	2.83%	4.41%	2.26%	1.87%	2.56%	1.94%
3	1	3.64%	2.95%	1.44%	1.21%	1.47%	1.28%
	5	2.78%	3.98%	2.35%	1.86%	2.41%	1.94%

Table IX: Updated NRMSEs for S4: STGP

Noise level	Clutter rate	Approach					
		Standard GP		DGPT-RBCM		DGPT-GPoE	
		X	Y	X	Y	X	Y
1	1	1.49%	0.96%	1.46%	0.94%	1.93%	1.11%
	5	2.77%	1.32%	2.75%	1.53%	2.99%	1.63%
2	1	1.48%	0.97%	1.47%	0.90%	1.93%	1.13%
	5	2.81%	1.30%	2.58%	1.49%	3.00%	1.61%
3	1	1.61%	0.99%	1.51%	0.97%	2.01%	1.12%
	5	2.81%	1.31%	2.64%	1.49%	2.94%	1.55%

corresponds to the 3σ confidence interval of the Gaussian distribution. Based on theoretical analysis from Theorem 2, the UCBs of predictions of DGPT-RBCM in both X and Y coordinates are presented in Figures 4 and 6. The confidence intervals of the predictive distribution of the DGPT-RBCM in both X and Y coordinates based on (19) and (20) are presented in Figures 5 and 7. The results in Figures 4 and 6 demonstrate that the proposed UCB can reveal the information of where the true location of the target is since the UCB can cover the true target location in most time steps with a high probability. However, in Figures 5 and 7, although the confidence intervals can quantify the uncertainties of the predictions themselves, the intervals fail to cover the true location of the target in most of the time steps. The derived UCB better characterizes the presence of the target in the error bound with 88% and 42% higher probability in X and Y coordinates, as compared to the confidence interval of DGP. The comparisons between the derived UCB and the confidence interval highlight the value and informativeness of the UCB, which can help to further refine the measurements for training the DGP model by excluding some measurements out of the bound and thus having the potential to further enhance the measurement preprocessing method presented in Section IV-C.

G. Hybrid Bayesian Filtering

In this section, the performance of the proposed hybrid Bayesian filtering approach for target tracking is evaluated as compared to both the Standard GP-based tracker and the DGPT approach. We have also compared with a

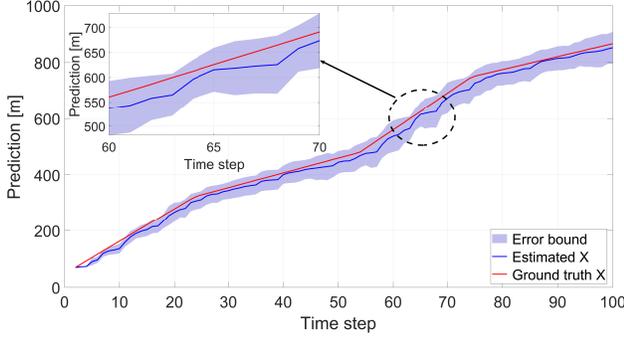


Figure 4: UCB for DGPT-RBCM of S1 in X coordinate

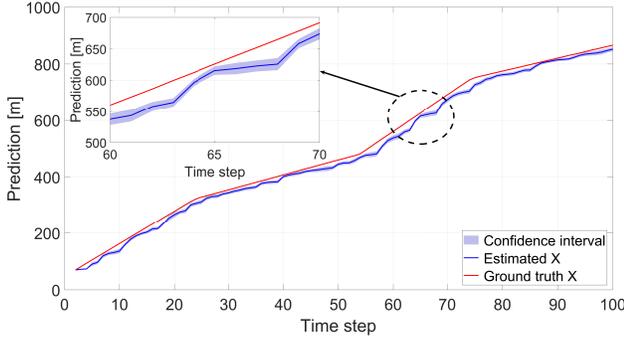


Figure 5: Confidence interval for DGPT-RBCM of S1 in X coordinate

model-based approach by using a convolutional particle filter (CPF) [68] (<https://doi.org/10.15131/shef.data.3543269.v2>) [66], [69]–[71] which is one of the best versions of the particle filters. The CPF is implemented by processing the data in a centralized manner.

The CPF has two types of models - one constant velocity model and 4 coordinated turn models. A grid of values for the angular turn rate ω is chosen as

$$\omega = \{0, 0.55, 0.49, -0.55, -0.49\} s^{-1}. \quad (53)$$

These values cover possible uniform motions and maneuver within the minimum and maximum values of the turn rate. The CPF multiple model filter is tested over all 4 considered scenarios, by keeping the same transition probabilities matrix, initial mode probabilities vector and grid for the angular turn values. This means that we give the same conditions for the CPF, for all testing scenarios.

Inspired by ideas from Gilholm and Salmond [61], [62], we adopt the powerful Poisson likelihood model in the CPF and in our proposed hybrid Bayesian filtering approach in order to deal with measurement origin uncertainties and solve the data association task.

The Poisson likelihood model for dealing with measurement origin uncertainties leads to one of the most efficient data association approaches which avoids combinatorial complexities which are typical for multiple hypothesis target tracking model-based approaches.

The dynamic model with a known turn rate has the follow-

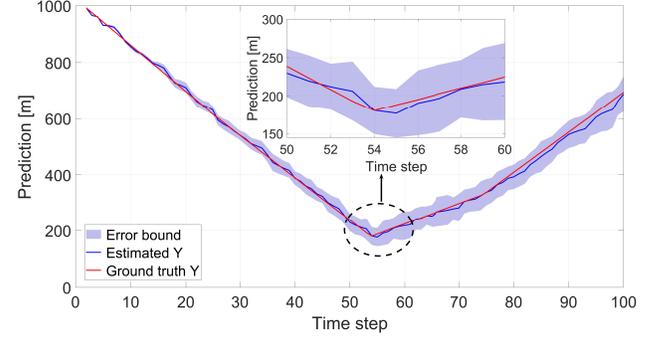


Figure 6: UCB for DGPT-RBCM of S1 in Y coordinate

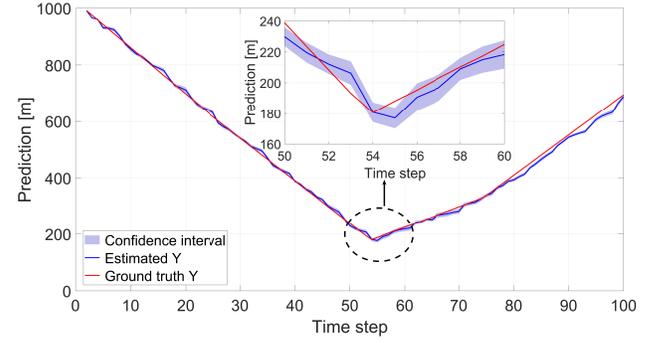


Figure 7: Confidence interval for DGPT-RBCM of S1 in Y coordinate

ing system matrix:

$$F(\omega) = \begin{bmatrix} 1 & \sin \omega T_s / \omega & 0 & -(1 - \cos \omega T_s) / \omega \\ 0 & \cos \omega T_s & 0 & -\sin \omega T_s \\ 0 & (1 - \cos \omega T_s) / \omega & 1 & \sin \omega T_s / \omega \\ 0 & \sin \omega T_s & 0 & \cos \omega T_s \end{bmatrix},$$

where T_s is the sampling interval and we set $T_s = 1$. The value of $\omega = 0$ corresponds to the nonmaneuvering (constant velocity) model. The values $\pm\omega$ correspond to the left and right turn, respectively. The transition mode probabilities are assumed to follow a Markov chain, with the initial mode probability as $\mu_0 = 0.6, \mu_1 = 0.1, \mu_2 = 0.1, \mu_3 = 0.1, \mu_4 = 0.1$. The mode transition matrix is defined as

$$\begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & P_{1,4} & P_{1,5} \\ P_{2,1} & P_{2,2} & P_{2,3} & P_{2,4} & P_{2,5} \\ P_{3,1} & P_{3,2} & P_{3,3} & P_{3,4} & P_{3,5} \\ P_{4,1} & P_{4,2} & P_{4,3} & P_{4,4} & P_{4,5} \\ P_{5,1} & P_{5,2} & P_{5,3} & P_{5,4} & P_{5,5} \end{bmatrix} = \begin{bmatrix} 0.84 & 0.05 & 0.03 & 0.05 & 0.03 \\ 0.05 & 0.88 & 0.07 & 0 & 0 \\ 0.03 & 0.07 & 0.9 & 0 & 0 \\ 0.05 & 0 & 0 & 0.88 & 0.07 \\ 0.03 & 0 & 0 & 0.07 & 0.9 \end{bmatrix}.$$

We choose the challenging case where the clutter rate is set as $\lambda_c = 5$ (high clutter case), and the measurement noise is set to be $\sigma_z^2 = 1$. The normalized root mean squared errors (NRMSEs) of the X and Y coordinates over 100 MC runs are

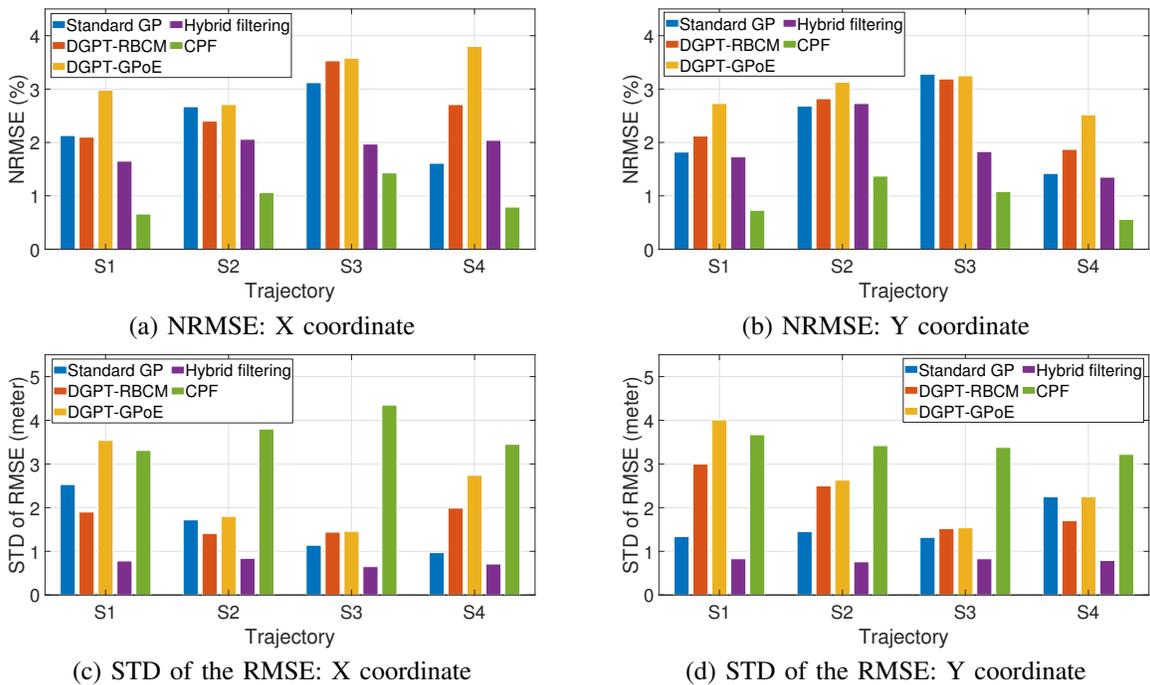


Figure 8: Tracking error and the STD of errors over 100 MC runs, $\sigma_z^2 = 1$, clutter rate=5

given in Figures A (a) and (b). The hybrid Bayesian filtering approach outperforms both the centralized and distributed GP-based tracking approaches by achieving the lowest NRMSEs of both coordinates (except the NRMSE in S4, X coordinate). In particular, we can see that a fine-tuned multiple-model CPF achieves the lowest NRMSE values in both X and Y coordinates.

We also present the standard deviation (STD) of averaged RMSEs over 100 MC runs. Each of the RMSEs is the tracking error calculated from a single independent MC run. The results are presented in Figures A (c) and (d). The proposed hybrid Bayesian filtering approach achieves the lowest STD of state estimations in both coordinates as compared to other GP-based approaches. It is also fairly stable over multiple scenarios. The results demonstrate that by involving the Poisson likelihood model, both the accuracy and the robustness of the proposed DGPT can be further improved. The STDs of RMSEs from the CPF are relatively higher in some cases, which shows the proposed algorithm achieves higher robustness than the model-based tracking approach. The high STDs of the RMSEs for the CPF also mean that the estimates from the CPF are somehow far from the true target trajectories in some areas or some runs.

VIII. CONCLUSION

In this paper, a novel DGP-based model-free learning and tracking approach is proposed to solve distributed point tracking problems in WSNs with clutter measurements. The developed distributed edge learning approach overcomes the limitations of standard GP-based tracking methods from a different angle via distributed GP. Theoretical derivations are presented for the UCB of the tracking error for two important tasks: 1)

when data have no clutter, 2) when the sensor data contain clutter which means there is measurement origin uncertainty. The UCBs characterize the trustworthiness of the proposed approach. The estimates are acceptable when the derived UCB is within certain pre-specified limits. The simulation results reveal that the proposed UCBs successfully encompass the true target states with 88% and 42% higher probability in X and Y coordinates, respectively, when compared to the confidence interval-based method. By introducing the Poisson measurement likelihood, a hybrid Bayesian filtering approach is proposed to merge the distributed machine learning and model-based methods to further improve the tracking performance and robustness. Numerical experiments demonstrate that the proposed approaches perform competitively well and can deal with varying motion models, noise levels, and clutter rates. Future work will focus on sensor management challenges in the developed distributed tracking system. The challenges include efficiently utilizing edge computing resources to accelerate DGP training and prediction. Another direction is to consider different types of measurements such as received signal strength and non-Gaussian measurement noises in the distributed tracking system. The theoretical derivation of a PCRLB is an important topic to focus on, and the validation of the theoretical results over real case studies will also be considered in the future.

APPENDIX A

According to (18), (19), Lemma 1 and the analysis as in Theorem 1, the deviation of the true function value from the

aggregated predictive mean by RBCM can be written as

$$|f(\mathbf{x}_*) - \mu_*^{\text{RBCM}}|, \\ = \left| f(\mathbf{x}_*) - \frac{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) \mu_i(\mathbf{x}_*)}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) + (1-M)\sigma_*^{-2}} \right|, \quad (54)$$

$$\approx \left| f(\mathbf{x}_*) - \frac{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*) \mu_i(\mathbf{x}_*)}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*)} \right|, \quad (55)$$

$$\leq \frac{\sum_{i=1}^M \beta_i \gamma_i^{1/2} \sigma_i^{-1}(\mathbf{x}_*)}{\sum_{i=1}^M \beta_i \sigma_i^{-2}(\mathbf{x}_*)}. \quad (56)$$

Equations (54)-(55) are derived when σ_*^{-2} (the variance of the prior distribution) has a big value which corresponds to lack of prior information or high uncertainty.

APPENDIX B

Based on Lemma 2, similar to Theorem 1, define B_i as the event that for any $t \in T$, the cumulative prediction of the target state from local expert i and the true target state differs larger than a quantity, which can be written as

$$B_i = \left\{ \bigcup_{t=1}^T |f(\mathbf{x}_t) - \mu^{\text{GPOE}}(\mathbf{x}_t)| > \gamma_{t,i}^{1/2} \sigma_i(\mathbf{x}_t) \right\}, \quad (57)$$

where $\mu_i(\mathbf{x}_t)$ and $\sigma_i(\mathbf{x}_t)$ represents the predictive mean and the STD of the function made by local expert i at time t , respectively.

Define the union of events $\{B_1, B_2, \dots, B_M\}$ as B . By applying the union bounds over M events, the probability of

B can be upper bounded as

$$\Pr(B) := \Pr \left\{ \bigcup_{i=1}^M B_i \right\} \leq \sum_{i=1}^M \delta_i. \quad (58)$$

Therefore, we have

$$\Pr(\bar{B}) := \Pr \left\{ \bigcap_{i=1}^M \bar{B}_i \right\}, \\ = \Pr \left\{ \bigcap_{i=1}^M \bigcap_{t=1}^T |f(\mathbf{x}_t) - \mu_i(\mathbf{x}_t)| \leq \gamma_{t,i}^{1/2} \sigma_i(\mathbf{x}_t) \right\}, \\ \geq 1 - \sum_{i=1}^M \delta_i. \quad (59)$$

Therefore, with probability $1 - \sum_{i=1}^M \delta_i$, we have

$$\sum_{t=1}^T |f(\mathbf{x}_t) - \mu_t^{\text{GPOE}}| \\ = \sum_{t=1}^T \left| f(\mathbf{x}_t) - \frac{\sum_{i=1}^M \beta_{t,i} \sigma_i^{-2}(\mathbf{x}_t) \mu_i(\mathbf{x}_t)}{\sum_{i=1}^M \beta_{t,i} \sigma_i^{-2}(\mathbf{x}_t)} \right|, \\ \leq \sum_{t=1}^T \frac{\sum_{i=1}^M \beta_{t,i} \sigma_i^{-2}(\mathbf{x}_t) |f(\mathbf{x}_t) - \mu_i(\mathbf{x}_t)|}{\sum_{i=1}^M \beta_{t,i} \sigma_i^{-2}(\mathbf{x}_t)}, \\ \leq \sum_{t=1}^T \frac{\sum_{i=1}^M \beta_{t,i} \gamma_{t,i}^{1/2} \sigma_i^{-1}(\mathbf{x}_t)}{\sum_{i=1}^M \beta_{t,i} \sigma_i^{-2}(\mathbf{x}_t)}. \quad (60)$$

APPENDIX C

The derivation of the expressions for the first two moments of the posterior state distribution is presented as follows:

$$p(\tilde{\mathbf{X}}_t | \mathbf{Z}_{t-C:t}) \propto p(\mathbf{z}_t | \tilde{\mathbf{X}}_t) p(\tilde{\mathbf{X}}_t | \mathbf{Z}_{t-C:t-1}), \quad (61)$$

$$= p(z_{t,1} | \tilde{\mathbf{X}}_t) p(z_{t,2} | \tilde{\mathbf{X}}_t) \cdots p(z_{t,n_t} | \tilde{\mathbf{X}}_t) p(\tilde{\mathbf{X}}_t | \mathbf{Z}_{t-C:t-1}) \quad (62)$$

$$= \prod_{j=1}^{n_t} \left\{ \frac{1}{\sqrt{2\pi\hat{\sigma}_t^2}} \exp \left(-\frac{(z_{t,j} - \hat{\mu}_t)^2}{2\hat{\sigma}_t^2} \right) \right\} \cdot \frac{1}{\sqrt{2\pi\sigma_{t-1}^2}} \exp \left(-\frac{(\tilde{x}_t - \mu_{t-1})^2}{2\sigma_{t-1}^2} \right), \quad (63)$$

$$\propto \exp \left\{ \frac{-\sum_{j=1}^{n_t} (z_{t,j}^2 - 2z_{t,j}\hat{\mu}_t + \hat{\mu}_t^2)}{2\hat{\sigma}_t^2} - \frac{\tilde{x}_t^2 - 2\tilde{x}_t\mu_{t-1} + \mu_{t-1}^2}{2\sigma_{t-1}^2} \right\}, \quad (64)$$

$$\propto \exp \left\{ \frac{-\sigma_{t-1}^2 \sum_{j=1}^{n_t} \left(z_{t,j}^2 - 2z_{t,j} \left(\frac{\lambda_c}{A_{\text{sen}}} + \lambda_T \tilde{x}_t \right) + \left(\frac{\lambda_c}{A_{\text{sen}}} + \lambda_T \tilde{x}_t \right)^2 \right) - \hat{\sigma}_t^2 (\tilde{x}_t^2 - 2\tilde{x}_t\mu_{t-1} + \mu_{t-1}^2)}{2\hat{\sigma}_t^2 \sigma_{t-1}^2} \right\}, \quad (65)$$

$$\propto \exp \left\{ \frac{-\tilde{x}_t^2 (\hat{\sigma}_t^2 + n_t \sigma_{t-1}^2 \lambda_T^2) + 2\tilde{x}_t \left(\mu_{t-1} \hat{\sigma}_t^2 + \sigma_{t-1}^2 \sum_{j=1}^{n_t} (z_j \lambda_T - \lambda_T \lambda_c / A_{\text{sen}}) \right)}{2\hat{\sigma}_t^2 \sigma_{t-1}^2} \right\}, \quad (66)$$

$$\propto \exp \left\{ \frac{-\tilde{x}_t^2 + 2\tilde{x}_t \left(\frac{\mu_{t-1} \hat{\sigma}_t^2 + \sigma_{t-1}^2 \sum_{j=1}^{n_t} (z_j \lambda_T - \lambda_T \lambda_c / A_{\text{sen}})}{\hat{\sigma}_t^2 + n_t \sigma_{t-1}^2 \lambda_T^2} \right)}{2 \frac{\hat{\sigma}_t^2 \sigma_{t-1}^2}{\hat{\sigma}_t^2 + n_t \sigma_{t-1}^2 \lambda_T^2}} \right\}, \quad (67)$$

$$\propto \exp \left\{ \frac{-\left(\tilde{x}_t - \frac{\mu_{t-1} \hat{\sigma}_t^2 + \sigma_{t-1}^2 \sum_{j=1}^{n_t} (z_j \lambda_T - \lambda_T \lambda_c / A_{\text{sen}})}{\hat{\sigma}_t^2 + n_t \sigma_{t-1}^2 \lambda_T^2} \right)^2}{2 \frac{\hat{\sigma}_t^2 \sigma_{t-1}^2}{\hat{\sigma}_t^2 + n_t \sigma_{t-1}^2 \lambda_T^2}} \right\}. \quad (68)$$

REFERENCES

- [1] É. L. Souza, E. F. Nakamura, and R. W. Pazzi, "Target tracking for sensor networks: A survey," *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–31, 2016.
- [2] L. Mihaylova, A. Y. Carmi, F. Septier, A. Gning, S. K. Pang, and S. Godsill, "Overview of Bayesian sequential Monte Carlo methods for group and extended object tracking," *Digital Signal Processing*, vol. 25, pp. 1–16, 2014.
- [3] Y. Bar-Shalom, P. Willett, and X. Tian, *Tracking and data fusion: A handbook of algorithms*. YBS Publishing: Storrs, CT, USA, 2011.
- [4] B. D. Anderson and J. B. Moore, *Optimal filtering*. Dover Publications, Inc., 2012.
- [5] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.
- [6] B. Ristic, D. E. Clark, and N. Gordon, "Calibration of multi-target tracking algorithms using non-cooperative targets," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 390–398, 2013.
- [7] B. Sobhani, E. Paolini, A. Giorgetti, M. Mazzotti, and M. Chiani, "Target tracking for UWB multistatic radar sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 1, pp. 125–136, 2014.
- [8] X. Zhang, P. Willett, and Y. Bar-Shalom, "Dynamic Cramér–Rao bound for target tracking in clutter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1154–1167, 2005.
- [9] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter: Particle filters for tracking applications*. Artech house, 2003.
- [10] M. L. Hernandez and A. Farina, "Posterior Cramér–Rao bound for target tracking in the presence of multipath," in *2018 21st International Conference on Information Fusion*, 2018, pp. 151–158.
- [11] Y. Zhao, C. Fritsche, G. Hendebay, F. Yin, T. Chen, and F. Gunnarsson, "Cramér–Rao bounds for filtering based on Gaussian process state-space models," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 5936–5951, 2019.
- [12] L. Mihaylova, D. Angelova, S. Honary, D. R. Bull, C. N. Canagarajah, and B. Ristic, "Mobility tracking in cellular networks using particle filtering," *IEEE Transactions on Wireless Communications*, vol. 6, no. 10, pp. 3589–3599, 2007.
- [13] X. Rong Li and V. Jilkov, "Survey of maneuvering target tracking. part V. multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.
- [14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [15] W. Aftab and L. Mihaylova, "A Gaussian process regression approach for point target tracking," in *Proceedings of the 22th International Conference on Information Fusion*, Ottawa, 2019, pp. 1–8.
- [16] —, "A learning Gaussian process approach for maneuvering target tracking and smoothing," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 57, no. 1, pp. 278–292, 2021.
- [17] —, "On the impact of different kernels and training data on a Gaussian process approach for target tracking," in *Proc. of the 23rd IEEE International Conference on Information Fusion*, Rustenburg, 2020, pp. 1–6.
- [18] M. Sun, M. E. Davies, I. Proudler, and J. R. Hopgood, "A Gaussian process based method for multiple model tracking," in *Proc. of the Sensor Signal Processing for Defence Conference*, 2020, pp. 1–5.
- [19] X. Lyu, B. Hu, K. Li, and L. Chang, "An adaptive and robust UKF approach based on Gaussian process regression-aided variational bayesian," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 9500–9514, 2021.
- [20] X. Liu, C. Lyu, J. George, T. Pham, and L. Mihaylova, "A learning distributed Gaussian process approach for target tracking over sensor networks," in *Proc. of the 25th International Conference on Information Fusion*, 2022, pp. 1–8.
- [21] C. Lyu, X. Liu, and L. Mihaylova, "Efficient factorisation-based Gaussian process approaches for online tracking," in *Proc. of the 25th International Conference on Information Fusion*, 2022, pp. 1–8.
- [22] W. Aftab, R. Hostettler, A. De Freitas, M. Arvaneh, and L. Mihaylova, "Spatio-temporal Gaussian process models for extended and group object tracking with irregular shapes," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2137–2151, 2019.
- [23] X. Tang, M. Li, R. Tharmarasa, and T. Kirubarajan, "Seamless tracking of apparent point and extended targets using Gaussian process PMHT," *IEEE Transactions on Signal Processing*, vol. 67, no. 18, pp. 4825–4838, 2019.
- [24] J. Peršić, L. Petrović, I. Marković, and I. Petrović, "Spatiotemporal multisensor calibration via Gaussian processes moving target tracking," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1401–1415, 2021.
- [25] M. Kok and A. Solin, "Scalable magnetic field slam in 3D using Gaussian process maps," in *Proc. of the 21st International Conference on Information Fusion*, 2018, pp. 1353–1360.
- [26] M. Kok, J. D. Hol, and T. B. Schön, "Indoor positioning using ultra-wideband and inertial measurements," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1293–1303, 2015.
- [27] K. O. A. Ali, N. Ilic, M. S. Stankovic, and S. S. Stankovic, "Consensus-based distributed adaptive target tracking in camera networks using integrated probabilistic data association," *EURASIP J. Adv. Signal Process.*, vol. 2018, p. 13, 2018.
- [28] F. Meyer, T. Kropfreiter, J. L. Williams, R. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message passing algorithms for scalable multitarget tracking," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 221–259, 2018.
- [29] S. Mehriizi, A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "A Bayesian Poisson–Gaussian process model for popularity learning in edge-caching networks," *IEEE Access*, vol. 7, pp. 92 341–92 354, 2019.
- [30] R. A. da Silva and N. L. da Fonseca, "Resource allocation mechanism for a fog-cloud infrastructure," in *Proceedings of the IEEE International Conference on Communications*. IEEE, 2018, pp. 1–6.
- [31] B. Da, Y.-S. Ong, A. Gupta, L. Feng, and H. Liu, "Fast transfer Gaussian process regression with large-scale sources," *Knowledge-Based Systems*, vol. 165, pp. 208–218, 2019.
- [32] M. G. S. Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, "Machine learning at the network edge: A survey," *ACM Comput. Surv.*, vol. 54, no. 8, oct 2021.
- [33] A. Albert and L. Imsland, "Survey: mobile sensor networks for target searching and tracking," *Cyber-Physical Systems*, vol. 4, no. 2, pp. 57–98, 2018.
- [34] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," *Advances in Neural Information Processing Systems*, vol. 18, p. 1257, 2006.
- [35] M. K. Titsias, "Variational model selection for sparse Gaussian process regression," *Report, University of Manchester, UK*, 2009.
- [36] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4405–4423, 2020.
- [37] G. E. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [38] Y. Cao and D. J. Fleet, "Generalized product of experts for automatic and principled fusion of Gaussian process predictions," *arXiv preprint arXiv:1410.7827*, 2014.
- [39] V. Tresp, "A Bayesian Committee Machine," *Neural Computation*, vol. 12, no. 11, pp. 2719–2741, 2000.
- [40] M. Deisenroth and J. W. Ng, "Distributed Gaussian processes," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, Lille, 2015, pp. 1481–1490.
- [41] S. Cohen, R. Mbuva, T. Marwala, and M. Deisenroth, "Healing products of Gaussian process experts," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2068–2077.
- [42] H. Liu, J. Cai, Y. Wang, and Y. S. Ong, "Generalized robust Bayesian committee machine for large-scale Gaussian process regression," in *International Conference on Machine Learning*, 2018, pp. 3131–3140.
- [43] F. Yin and F. Gunnarsson, "Distributed recursive Gaussian processes for RSS map applied to target tracking," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 492–503, 2017.
- [44] M. Lázaro-Gredilla, S. Van Vaerenbergh, and N. D. Lawrence, "Overlapping mixtures of Gaussian processes for the data association problem," *Pattern recognition*, vol. 45, no. 4, pp. 1386–1395, 2012.
- [45] M. Kaiser, C. Otte, T. A. Runkler, and C. H. Ek, "Data association with Gaussian processes," in *Lecture Notes in Computer Science, from the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 548–564.
- [46] J. Hartikainen and S. Särkkä, "Kalman filtering and smoothing solutions to temporal Gaussian process regression models," in *In Proc. of the IEEE International Workshop on Machine Learning for Signal Processing*, 2010, pp. 379–384.
- [47] S. Sarkka, A. Solin, and J. Hartikainen, "Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing: A look at Gaussian process regression through Kalman filtering," *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 51–61, 2013.
- [48] H. Nickisch, A. Solin, and A. Grigorevskiy, "State Space Gaussian Processes with Non-Gaussian Likelihood," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80, 10–15 Jul 2018, pp. 3789–3798.
- [49] R. Turner, M. Deisenroth, and C. Rasmussen, "State-space inference and learning with Gaussian processes," in *Proceedings of the Thirteenth*

- International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings*, 2010, pp. 868–875.
- [50] C. Veibäck and J. Olofsson and T. Lauknes and G. Hendeby, “Learning target dynamics while tracking using Gaussian processes,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 4, pp. 2591–2602, 2020.
- [51] A. Kullberg, I. Skog, and G. Hendeby, “Online joint state inference and learning of partially unknown state-space models,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 4149–4161, 2021.
- [52] N. Ilić, M. S. Stanković, and S. S. Stanković, “Adaptive consensus-based distributed target tracking in sensor networks with limited sensing range,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 2, pp. 778–785, 2014.
- [53] J. George, “Distributed tracking of maneuvering target: A finite-time algorithm,” in *Proc. of the 19th IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 4420–4424.
- [54] M. Raitoharju, Á. F. García-Fernández, R. Hostettler, R. Piché, and S. Särkkä, “Gaussian mixture models for signal mapping and positioning,” *Signal Processing*, vol. 168, p. 107330, 2020.
- [55] R. B. Gramacy, “laGP: large-scale spatial modeling via local approximate Gaussian processes in R,” *Journal of Statistical Software*, vol. 72, pp. 1–46, 2016.
- [56] S. E. Yuksele, J. N. Wilson, and P. D. Gader, “Twenty years of mixture of experts,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1177–1193, 2012.
- [57] S. Wang, Y. Hong, R. Wang, Q. Hao, Y.-C. Wu, and D. W. K. Ng, “Edge federated learning via unit-modulus over-the-air computation,” *IEEE Transactions on Communications*, vol. 70, no. 5, pp. 3141–3156, 2022.
- [58] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, “Federated learning: A signal processing perspective,” *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 14–41, 2022.
- [59] K. Berntorp, “Online Bayesian inference and learning of Gaussian-process state-space models,” *Automatica*, vol. 129, p. 109613, 2021.
- [60] N. Srinivas, A. Krause, S. Kakade, and M. Seeger, “Gaussian process optimization in the bandit setting: No regret and experimental design,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Madison, WI, USA, 2010, p. 1015–1022.
- [61] K. Gilholm, S. Godsill, S. Maskell, and D. Salmond, “Poisson models for extended target and group tracking,” in *Proceedings of Signal and Data Processing of Small Targets Conference*, vol. 5913. SPIE, 2005, pp. 230 – 241.
- [62] K. Gilholm and D. Salmond, “Spatial distribution model for tracking extended objects,” *IEE Proceedings-Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364–371, 2005.
- [63] M. Canaud, L. Mihaylova, J. Sau, and N.-E. El Faouzi, “Probability hypothesis density filtering for real-time traffic state estimation and prediction,” *Network and Heterogeneous Media*, vol. 8, no. 3, pp. 825–842, Sep. 2013.
- [64] M. Lundgren, L. Svensson, and L. Hammarstrand, “A CPHD filter for tracking with spawning models,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 496–507, 2013.
- [65] C. Lundquist, K. Granström, and U. Orguner, “An extended target CPHD filter and a Gamma Gaussian inverse wishart implementation,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 472–483, 2013.
- [66] A. De Freitas, L. Mihaylova, A. Gning, D. Angelova, and V. Kadiramanathan, “Autonomous crowds tracking with box particle filtering and convolution particle filtering,” *Automatica*, vol. 69, pp. 380–394, 2016.
- [67] A. Gning and L. Mihaylova, “Dynamic clustering and belief propagation for distributed inference in random sensor networks with deficient links,” in *Proc. of the 12th International Conference on Information Fusion*, Seattle, 2009, pp. 656–663.
- [68] D. Angelova, L. Mihaylova, N. Petrov, and A. Gning, “A convolution particle filtering approach for tracking elliptical extended objects,” in *Proceedings of the 16th International Conference on Information Fusion*, 2013, pp. 1542–1549.
- [69] A. Doucet, N. de Freitas, and N. Gordon, *An Introduction to Sequential Monte Carlo Methods*. New York, NY: Springer New York, 2001, pp. 3–14.
- [70] C. Musso, N. Oudjane, and F. Le Gland, “Improving regularised particle filters in sequential Monte Carlo methods in practice, Eds: A. Doucet, N. de Freitas, N. Gordon, Statistics for engineering and information science,” 2001.
- [71] D. Angelova, L. Mihaylova, N. Petrov, and A. Gning, “Extended object tracking with convolution particle filtering,” in *Proceedings of the 6th IEEE International Conference Intelligent Systems*, 2012, pp. 96–101.